Infarct – analiza si predictii

Fiind o persoana cu un stil de viata nesanatos, am decis sa fac acest studiu cu scopul de a imi da mie un semnal de alarma dar si celorlalti. Prin acest studiu ne dorim sa raspundem la intrebarea:”Putem calcula probabilitatea unui pacient sa faca infarct?”. Exista insa si alte intrebari la care am putea primi raspuns, cum ar fi cardrele sociale din care fac parte persoanele predispuse sa faca infarct(stare civila, tipul zonei in care locuieste).

Aceste intrebari sunt relevante pentru cei preocupati de sanatatea lor, dar si pentru doctori, farmacisti, guvern. Avand raspunsul la aceste intrebari am putea schimba sau introduce in sistemul de invatamant un subiect nou, cu scopul de a aduce informatia si constientizarea in randul tuturor.

Setul de date, luat de pe kaggle.com (incarcat de utilizatorul **FEDESORIANO)** folosit contine 12 coloane si 5110 inregistrari:

1) id: identificator unic  
2) gender: "Male", "Female" sau "Other"  
3) age: varsta pacientului  
4) hypertension: 0 daca nu are probleme cu tensiunea, 1 alfel  
5) heart\_disease: 0 daca pacientul nu are boli de inima, 1 daca are  
6) ever\_married: "No" sau "Yes"  
7) work\_type: "children", "Govt\_jov", "Never\_worked", "Private" sau "Self-employed"  
8) Residence\_type: "Rural" sau "Urban"  
9) avg\_glucose\_level: nivelul mediu de glucoza in sange  
10) bmi: indicele de masa corporala  
11) smoking\_status: "formerly smoked", "never smoked", "smokes" sau "Unknown"  
12) stroke: 1 daca pacientul a facut infarct, 0 altfel

Nota: "Unknown" in smoking\_status inseamna ca informatia nu este disponibila pentru acest pacient

Pentru realizarea acestui studiu se va folosi python, un limbaj interpretabil, fiind unul dintre cele mai populare limbaje, acesta este echipat cu o multitudine de librarii care servesc diverselor domenii, astfel ca nu este nevoie sa se faca calcule scriind intr-un limbaj low level.

Toate coloanele sunt populate in intregime, exceptand coloana „bmi”, care contine unele valori de tip NULL( numpy.NAN in python). Am putea scapa de acestea, nefiind relevante pentru studiu, insa inainte de asta trebuie sa verificam daca exista pacienti care au facut infarct printre acestia. Se constata faptul ca exista suficienti pacienti printre acestia incat sa nu putem sa stergem inregistrarile, astfel ca vom schimba valorile null cu media coloanei bmi, insa nu inainte sa modificam valorile mai mari de 40 cu 40, 40 fiind o valoare a indicelui la care pacientul sufera de obezitate morbida. In urma acestei operatiuni media scade putin ( de la 28.893237 la 28.422963), lucru care nu afecteaza studiul.

Chart, bar chart

Description automatically generated

Putem observa ca raportul dintre persoanele care au facut infarct si cele care nu este foarte mare, 4.9% din pacientii mentionati in acest dataset au suferit de infact. In acelasi timp faptul ca sunt doar 5110 de inregistrari nu ne ofera o acuratete sufient de mare. Aceasta problema se poate aborda in doua feluri:

1. Selectam datele care au stroke = 1 impreuna cu o parte (aleatorie) din celelalte date( 1000-1400)
2. Folosim metoda F1 score, care presupune folosirea unui scor pentru a determina algoritmul potrivit.

Daca acuratete = #predictii corecte / #total predictii, in metodolofia F1 avem:  
 acuratete = # cazuri pozitive / # cazuri pozitive + #fals pozitive

Si se introduce si notiunea de „rechemare” sau „reamintire” si are formula:

R = #cazuri pozitive / #cazuri pozitive + #fals negative

Iar scorul F1 se calculeaza : 2 \* (acuratete \* rechemare) / (acuratete + rechemare)

Chart, bar chart

Description automatically generated

Explorand datele, putem observa ca nu poate fi definitoriu faptul ca o persoana este casatorita sau nu in a determina daca un pacient e predispus sa faca infarct.

Chart, bar chart

Description automatically generated

Sexul nu este nici el definitoriu, desi inclin spre a spune ca barbatii sunt mai predispusi sa faca infact, considerand diagrama de mai sus. Observam de asemenea ca avem 1 other gender, neavand sens sa avem inca o coloana pentru un singur record, vom inlocui acea valoare cu un standard binar

Chart, bar chart

Description automatically generated

Din diagrama de mai sus putem trage concluzia ca ce cei care au muncit au sansele mai mari sa faca infarct, comparativ cu cei care nu au muncit sau au muncit mai putin.

Chart, bar chart

Description automatically generated

Similar, putem trage concluzia ca pacientii diagnosticati cu hipertensiune sunt mai predispusi sa faca infarct, insa nu este un element definitoriu.

Chart, bar chart

Description automatically generated

Faptul ca pacientul provine dintr-un mediu rural sau urban nu este definitoriu si nu pare sa influenteze probabilitatea unui pacient de a suferi de infarct.

Chart, histogram

Description automatically generated

Diagrama de mai sus ne ajuta sa tragem concluzia ca varsta e un element foarte important in a determina predispozitia de a suferi de infarct, observandu-se ca majoritatea au avut loc la pacientii aflati la varsta de 80 de ani.

Chart, scatter chart

Description automatically generated

De aici putem observa ca pacientii inaintati in varsta sunt mai predispusi la diabet, pacientii inaintati in varsta sunt predispusi sa sufere de infarct, pacientii cu diabet(nivel inalt al glucozei in sange) sunt predispusi sa sufere de infarct.

Chart, bar chart

Description automatically generated

Neavand toate datele, nu putem trage concluzii singure, insa se poate observa o predispozitie a fumatorilor si fostilor fumatori sa sufere de infarct.

Urmatorul pas este sa transformam datele care nu sunt numerice in date numerice, pentru asta python vine in ajutor cu o librarie „sklearn”, aceeasi librarie cu care ulterior vom putea scala datele, astfel ca vom transforma coloanele gender, ever\_married, work\_type, residence\_type, smoking\_status in coloane cu valori numerice.

A picture containing text, monitor, road, screen

Description automatically generated

Generand un heatmap vom putea observa faptul ca singurele coloane corelate intre ele sunt age si ever\_married. Astfel ca putem renunta la ever\_married, fara sa afecteze rezultatul finalul. Pentru moment am ales sa nu fac asta, in acelasi timp consider ca orice tip de informatie este foarte utila si ar putea fi un caz diferential pentru multe persoane

A screenshot of a computer

Description automatically generated with medium confidence

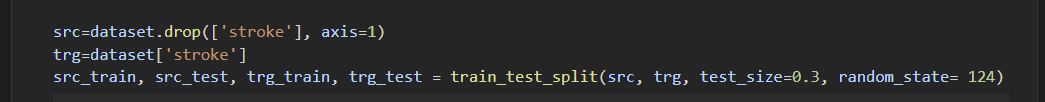
Acum ca am transformat valorile text in valori numerice, observam si faptul ca acestea contribuie diferit, fiind calculate la scara diferita. Nu putem avea rezultate valide daca nu le scalam, astfel incat acestea sa fie in acelasi range. In urma scalarii rezultatele arata asa:

A screenshot of a computer

Description automatically generated with low confidence

Urmeaza testarea mai multori algoritmi (Arborele de decizii, regresia logstica, random forest, support vector machine, naive bytes, cel mai apropiat vecin (KNN), naive bayes si Kmeans clustering). Pentru a lucra cu ele vom folosi tot libraria sklearn.

Vom incepe prin a initializa trainerul



In source avem inputul care este reprezentat de toata baza de date, mai putin coloana stroke, iar ca target avem coloana stroke. Test\_size reprezinta valoarea de referinta pentru impartirea subdatelor(poate fi de la 0 la 1 si reprezinta in procente cate recorduri sa fie considerate la un test, pentru inceput vom merge cu 30%, poate fi integer ceea ce specifice valoarea absoluta de recorduri care ar trebui considerate ex: 500) . Random\_state controleaza randomizarea datelor

------------------------------Logisticreg------------------------------

[[1436 8]

[ 88 1]]

precision recall f1-score support

0 0.94 0.99 0.97 1444

1 0.11 0.01 0.02 89

accuracy 0.94 1533

macro avg 0.53 0.50 0.49 1533

weighted avg 0.89 0.94 0.91 1533

------------------------------DecisionTree------------------------------

[[1378 66]

[ 71 18]]

precision recall f1-score support

0 0.95 0.95 0.95 1444

1 0.21 0.20 0.21 89

accuracy 0.91 1533

macro avg 0.58 0.58 0.58 1533

weighted avg 0.91 0.91 0.91 1533

------------------------------kNN------------------------------

[[1444 0]

[ 89 0]]

precision recall f1-score support

0 0.94 1.00 0.97 1444

1 0.00 0.00 0.00 89

accuracy 0.94 1533

macro avg 0.47 0.50 0.49 1533

weighted avg 0.89 0.94 0.91 1533

------------------------------NaiveBayes------------------------------

[[1437 7]

[ 82 7]]

precision recall f1-score support

0 0.95 1.00 0.97 1444

1 0.50 0.08 0.14 89

accuracy 0.94 1533

macro avg 0.72 0.54 0.55 1533

weighted avg 0.92 0.94 0.92 1533

------------------------------SupportVectorMachine------------------------------

[[1444 0]

[ 89 0]]

precision recall f1-score support

0 0.94 1.00 0.97 1444

1 0.00 0.00 0.00 89

accuracy 0.94 1533

macro avg 0.47 0.50 0.49 1533

weighted avg 0.89 0.94 0.91 1533

------------------------------RandomForest------------------------------

[[1442 2]

[ 89 0]]

precision recall f1-score support

0 0.94 1.00 0.97 1444

1 0.00 0.00 0.00 89

accuracy 0.94 1533

macro avg 0.47 0.50 0.48 1533

weighted avg 0.89 0.94 0.91 1533

------------------------------KMeans------------------------------

[[688 756]

[ 44 45]]

precision recall f1-score support

0 0.94 0.48 0.63 1444

1 0.06 0.51 0.10 89

accuracy 0.48 1533

macro avg 0.50 0.49 0.37 1533

weighted avg 0.89 0.48 0.60 1533

Raportul de mai sus este rezultatul executiei celor 7 algoritmi. Cel mai important este ca in coloana de recall sa existe pentru stroke (1) un rezultat cat mai mare, altfel exista predispozitia unor fals negative, de asemenea nu dorim nici fals pozitive.

Un raport al acuratetii:

Text

Description automatically generated

Din pacate nu vom putea folosi kmeans(arele cele mai mari valori echilibrate la recall), deoarece are o acuratete mult prea scazuta.

Crescand train\_size la 50%, scanzand apoi la 25%, ulterior la 10%, s-a putut observa cum Modelul arborelui de decizii ar fi potrivit pentru acest studiu, insa valorile din recall sunt inca mult prea mici.

Graphical user interface

Description automatically generated with low confidence

Renuntand la coloana „ever\_married” (care este corelata cu varsta), si la coloana ID de care nu aveam nevoie am reusit sa obtin rezultate mai bune cu un test size de 30% si random state de 124 cu modelul naive bayes

Graphical user interface

Description automatically generated with low confidence

Celelalte test size-uri si random state testate avand in jur de 0.31 recall

Pentru a valida aceste rezultate putem folosi cross\_val\_score din modulul sklearn

Ruland acest cross validation cu un fold de 100, media este de 0.8730, ceea ce inseamna ca avem o acuratete de 87.30% in medie.

Studii similare au reusit sa atinga o acuratete de 99%, eliminand o parte random din datele persoanelor care nu au avut infarct. Lucru menit sa dea un echilibru datasetului.

In concluzie Naive Bayes este modelul cel mai bun pentru acest studiu, data fiind performanta si acuratetea acestuia in testele facute, in acelasi timp consider ca daca datasetul era mai mare, s-ar fi putut atinge o acuratete mult mai mare, sau daca aveam situatia fumatorilor exacta.